

Analisis Performa Rasio Kompresi Pada Metode Differensiasi ASCII Dan Lempel Ziv Welch (LZW)

Tommy¹, Rosyidah Siregar², Amir Mahmud Husein³, Mawaddah Harahap⁴, Ferdy Riza⁵

Address: Universitas Harapan Medan, Fakultas Teknologi Dan Komputer, Program Studi Teknik Informatika, Indonesia.^{1,2} Universitas Prima Indonesia, Fakultas Teknologi Dan Ilmu Komputer, Program Studi Teknik Informatika, Indonesia.^{3,4} STMIK Logika, Program Studi Teknik Informatika, Indonesia.⁵

Email: tommy.unhar@harapan.ac.id¹, rosyidah_siregar.unhar@harapan.ac.id², amirmahmud@unprimdn.ac.id³, mawaddah@unprimdn.ac.id⁴, ferdyriza@logika.ac.id⁵

Abstrak

Differensiasi ASCII merupakan salah satu metode kompresi yang memanfaatkan nilai differensi atau selisih antar byte yang terdapat pada karakter masukan. Secara teknis metode differensiasi ASCII dapat dilakukan dengan menggunakan kamus pengkodean ataupun menggunakan *windowing block* pengganti kamus pengkodean. Penelitian terdahulu yang telah dilakukan menunjukkan rasio kompresi differensi ASCII cukup baik namun masih perlu dilakukan analisis terhadap performa dari sudut pandang rasio kompresi metode tersebut dibandingkan dengan metode lainnya yang telah banyak digunakan saat ini. Pada penelitian ini akan dilakukan analisis terhadap perbandingan metode Differensi ASCII dengan metode kompresi lainnya seperti LZW. Pemilihan LZW sendiri dilakukan dengan alasan banyaknya aplikasi kompresi data yang menggunakan metode tersebut sehingga dapat menjadi tolak ukur yang tepat. Perbandingan rasio kompresi yang dilakukan menunjukkan hasil differensi ASCII memiliki keunggulan dibandingkan dengan LZW khususnya pada karakter masukan yang berukuran kecil. Sedangkan pada karakter masukan yang berukuran besar, LZW dapat mengoptimalkan probabilitas pasangan karakter yang muncul dibandingkan dengan differensiasi ASCII yang terpaku pada nilai selisih pada setiap blok karakter masukan sehingga pada karakter dengan ukuran yang besar LZW memiliki rasio kompresi yang lebih besar dibandingkan dengan differensiasi ASCII.

Keywords – Kompresi, Rasio, Differensiasi ASCII, Lempel Ziv Welch

1. Latar Belakang

Differensiasi ASCII merupakan hasil pengembangan metode kompresi *delta encoding* yang memanfaatkan nilai selisih byte dari simbol atau karakter masukan [1],[2]. Model kompresi differensiasi ASCII menggunakan *windowingblock* atau blok – blok berukuran tetap dan menghitung nilai selisih dari byte – byte karakter yang terdapat pada setiap blok untuk menjaga nilai hasil selisih agar tidak terlalu besar sehingga rasio kompresi dapat tetap terjaga [3]. Dengan memanfaatkan nilai selisih sebagai pengkodean dari karakter masukan, maka hasil kompresi akan menjadi lebih sederhana dimana nilai selisih biasanya memiliki panjang bit yang lebih kecil dibandingkan dengan byte karakter biasa.

Rasio kompresi yang cukup baik dan stabil pada metode differensiasi ASCII membutuhkan analisis lebih lanjut dengan membandingkannya dengan metode kompresi

lainnya. Dimana dengan melakukan perbandingan dengan metode kompresi lainnya dapat diperoleh keunggulan dan kekurangan antara satu metode dengan metode lainnya.

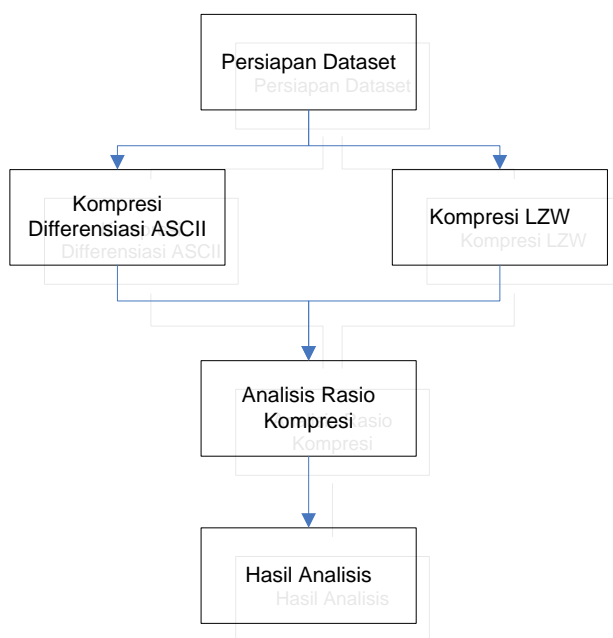
Perbandingan antara metode kompresi biasanya dilakukan dengan membandingkan rasio kompresi yang dihasilkan oleh masing – masing metode kompresi. Rasio kompresi dapat digunakan untuk menunjukkan efisiensi sebuah metode dibandingkan dengan metode lainnya [4]. Perbandingan dari berbagai metode kompresi telah dilakukan sebelumnya pada beberapa penelitian [5]-[10]. Sebagian besar penelitian menggunakan masukan teks sebagai perbandingan dikarenakan hampir semua metode kompresi dapat bekerja dengan baik dengan masukan teks. Pada penelitian terbaru perbandingan telah berkembang dengan menggunakan berbagai jenis masukan yang tidak hanya teks saja namun juga

menggunakan masukan lain seperti suara, video, citra digital, dokumen khusus dan lain sebagainya [11]-[15].

Pada penelitian ini akan dilakukan perbandingan metode kompresi dengan metode kompresi lainnya. Salah satu metode kompresi yang memiliki rasio kompresi yang cukup baik dan banyak digunakan saat ini adalah metode LZW. Metode LZW telah banyak digunakan pada beberapa penelitian perbandingan kompresi [4]-[11], [13],[16]. Sebagian hasil penelitian perbandingan menunjukkan LZW memiliki hasil rasio kompresi yang cukup baik sehingga pada penelitian ini juga akan digunakan sebagai pembanding terhadap metode kompresi differensiasi ASCII.

2. Metode

Metode yang digunakan pada penelitian ini adalah metode kompresi differensiasi ASCII dan metode kompresi LZW yang akan dibandingkan untuk memperoleh hasil analisis perbandingan rasio kedua metode tersebut pada beberapa masukan yang berbeda seperti text, citra digital, dokumen, dan masukan lainnya. Adapun langkah penelitian yang dilakukan pada penelitian ini dapat dilihat pada gambar 1.



Gambar 1 Tahapan Proses Perbandingan

2.1. Differensiasi ASCII

Model metode differensiasi ASCII yang digunakan pada penelitian ini adalah model yang diajukan pada artikel penelitian sebelumnya [3]. Pada metode differensiasi ASCII *Encoding* dilakukan dengan menghitung nilai numeric terendah dan tertinggi dari karakter yang

terdapat pada pesan teks menggunakan tabel ASCII. Selanjutnya akan dihitung diferensi dari nilai terendah dan tertinggi tersebut. Nilai diferensi tersebut akan dibagi dua untuk mencari nilai tengah dari nilai diferensi tersebut. Nilai tengah tersebut akan menjadi nilai titik referensi atau *reference point*. Proses *encoding* kemudian dilakukan dengan melakukan substitusi nilai diferensi yang diperoleh dari tiap karakter terhadap titik referensi.

Berikut tahapan *encoding* dari differensiasi ASCII :

1. Mencari karakter dengan nilai kode ASCII terendah dan tertinggi.
2. Menghitung nilai diferensi dari karakter terendah dan tertinggi tersebut :

$$d = \text{MaxC} - \text{MinC}$$

Dimana :

d = nilai diferensi.

MinC = Nilai kode ASCII terendah dari karakter pesan

MaxC = Nilai kode ASCII tertinggi dari karakter pesan

3. Menghitung nilai tengah dari nilai diferensi dengan menggunakan pembulatan keatas :

$$\text{Mid} = d/2$$

4. Mencari nilai titik referensi :
$$p = \text{MinC} + \text{Mid}$$
5. Menghitung nilai diferensi tiap karakter terhadap nilai titik referensi.
6. Melakukan *encoding* karakter ditambahkan dengan nilai titik referensi sebagai byte awal.

Decoding dilakukan dengan menghitung nilai ASCII karakter asal menggunakan diferensi antara nilai diferensi karakter yang terkompresi terhadap nilai titik referensi.

Berikut tahapan proses *decoding* :

1. Inisialisasi Titik Referensi.
2. Menghitung selisih tiap bit karakter ter-*encode* dengan nilai titik referensi.
3. Konversi nilai diferensi ke karakter ASCII.

Pada differensiasi ASCII dasar, digunakan nilai acuan atau referensi dari titik tengah antara nilai karakter terendah dengan nilai karakter tertinggi, sehingga setiap karakter akan disubstitusikan dengan bit dari nilai differensiasi secara global sehingga semakin besar nilai differensiasi maka hasil kompresi memiliki rasio kompresi yang buruk bahkan tidak terkompresi sama sekali.

1. Mengambil beberapa karakter yang terdapat pada area *window*.

2. Menghitung nilai diferensi dari karakter terendah dan tertinggi dari window tersebut :

$$d = \text{MaxC} - \text{MinC}$$

Dimana :

d = nilai diferensi.

MinC = Nilai kode ASCII terendah dari karakter pesan

MaxC = Nilai kode ASCII tertinggi dari karakter pesan

3. Jika nilai $d \leq 15$ maka ukuran window dapat diperbesar sesuai kebutuhan selama $d \leq 15$ yang mana harapannya adalah bit hasil encode tidak lebih dari empat bit.

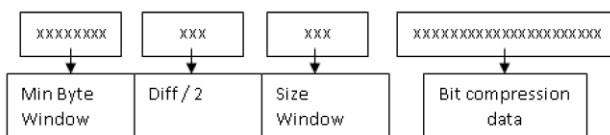
4. Melakukan encoding terhadap kode ASCII dari karakter berdasarkan nilai differensinya terhadap nilai titik referensi.

5. Menyusun bit header hasil kompresi agar dapat di dekompresi kembali:

Header :

Bit pertama sampai bit ke - n = informasi window

Bit n+1 sampai end = bit hasil encoding



Gambar 2. Struktur Blok Hasil Encoding Differensiasi ASCII

2.2. Lempel Ziv Welch(LZW)

Lempel Ziv Welch merupakan versi pengembangan dari algoritma LZ78. Dipublikasikan oleh Terry Welch pada tahun 1984, yang secara dasar menggunakan prinsip dari algoritma LZSS. Lempel Ziv Welch mengompresi data menjadi hampir dari separuh dari ukuran sebenarnya. Lempel Ziv Welch juga memberikan hasil yang cukup baik terhadap data yang dipenuhi dengan banyak data redundan seperti tabulasi angka, source code, dan data sinyal [15].

Secara umum proses encoding pada algoritma Lempel Ziv Welch dapat dijabarkan sebagai berikut [16].

- a. Inisialisasi kamus yang digunakan dengan symbol pertama dari deretan symbol pada pesan.
- b. Temukan entri terpanjang W pada kamus yang sesuai dengan input I .
- c. Tambahkan indeks dari entri yang sesuai pada output dan hilangkan W dari I .

- d. Tambahkan sebuah entri baru Wx ke dalam kamus, dimana x adalah symbol berikutnya yang akan dienkoding.
- e. Ulangi langkah 2 sampai proses encoding selesai.

3. Hasil

Pengujian perbandingan yang dilakukan pada penelitian ini menggunakan dataset dengan jenis atau format *file* yang berbeda. Format berkas atau *file* yang digunakan pada penelitian ini adalah format text (txt, html, xml, doc), citra digital (jpg, png, bmp), audio (wma dan mp3), video (avi, wmv dan 3gp). Adapun tabel dataset yang digunakan pada penelitian ini dapat dilihat pada tabel 1. Setiap jenis format *file* akan dikompresi secara terpisah antara kompresi differensiasi ASCII dan kompresi LZW. Analisis rasio kompresi antara metode differensiasi ASCII dan metode LZW akan membahas perbandingan hasil rasio kompresi dari tiap jenis format *file* dan berdasarkan ukuran dari *file* tersebut.

Tabel 1. Dataset *File* Teks

Nama File	Ukuran File (Byte)
Text1.txt	46
Text2.txt	11161
Text3.txt	38002
Text4.txt	44637
Dokumen1.docx	15137
Dokumen2.docx	207652
Dokumen3.docx	1528983
wowslider.html	1772
wowslider-howto.html	6457
wowslider-iframe.html	247

Tabel 2. Dataset *File* Multimedia

Nama File	Ukuran File (Byte)
Gambar1.jpg	12352
Gambar2.png	270054
Gambar3.png	8177
Audio1.mp3	1622673
Audio2.mp3	2001344
Audio3.wma	140547
Video1.avi	837120
Video3.3gp	344169
Video2.avi	675840

Hasil perbandingan kompresi pada metode differensiasi ASCII dan metode kompresi LZW pada masukan tipe teks memiliki hasil kompresi yang sangat variatif yang mana

dapat dilihat pada tabel 3. Sedangkan rasio kompresi dari hasil kompresi kedua metode dapat dilihat pada tabel4.

Tabel 3. Perbandingan Hasil Kompresi Teks

Nama File	Ukuran Awal	Kompresi	
		DiffASCII	LZW
Text1.txt	46	33	45
Text2.txt	11161	7047	3182
Text3.txt	38002	22738	11363
Text4.txt	44637	34982	13208
Dokumen1.docx	15137	13092	18532
Dokumen2.docx	207652	202824	252013
Dokumen3.docx	1528983	1503670	1759160
wowslider.html	1772	1406	1100
wowslider-howto.html	6457	5000	3278
wowslider-iframe.html	247	189	213

Tabel 4. Perbandingan Rasio Hasil Kompresi Teks

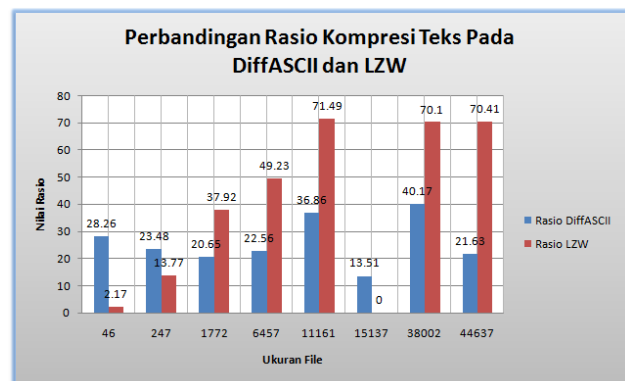
Nama File	Ukuran Awal	Rasio Kompresi	
		DiffASCII	LZW
Text1.txt	46	28.26	2.17
Text2.txt	11161	36.86	71.49
Text3.txt	38002	40.17	70.1
Text4.txt	44637	21.63	70.41
Dokumen1.docx	15137	13.51	0
Dokumen2.docx	207652	2.33	0
Dokumen3.docx	1528983	1.66	0
wowslider.html	1772	20.65	37.92
wowslider-howto.html	6457	22.56	49.23
wowslider-iframe.html	247	23.48	13.77

Pada tabel 2 dapat dilihat bahwa pada kompresi differensiasi ASCII setiap masukan berkas teks dapat dikompresi menjadi lebih kecil. Sedangkan kompresi LZW pada beberapa kasus berkas khususnya berkas DocX, hasil kompresi cenderung menghasilkan hasil keluaran dengan ukuran yang lebih besar dibandingkan dengan ukuran aslinya.

Rasio kompresi dari kedua metode yang dapat dilihat pada tabel 3 menunjukkan rasio kompresi tertinggi yang mampu dicapai oleh differensiasi ASCII hanyalah 40.17 % dibandingkan dengan metode LZW yang mampu

mencapai 71.49 %. Namun, metode differensiasi ASCII selalu berhasil melakukan kompresi pada semua berkas masukkan dimana rasio kompresi selalu berada diatas 0 sedangkan LZW yang mengalami kegagalan pada kompresi beberapa file teks memiliki rasio kompresi yang tidak bernilai sama sekali dikarenakan hasil kompresi justru menghasilkan keluaran yang lebih besar ukurannya.

Secara garis besar dapat dilihat bahwa pada masukan teks, metode kompresi differensiasi ASCII memiliki rasio kompresi yang lebih stabil dibandingkan dengan metode LZW namun sulit untuk mencapai nilai rasio kompresi yang tinggi dibandingkan dengan LZW.



Gambar 3. Grafik Perbandingan Rasio Kompresi Teks
Perbandingan rasio kompresi kemudian dilanjutkan pada masukan multimedia yang terdiri dari jenis file citra digital, suara dan video yang mana dapat dilihat pada tabel 5 dan 6 berikut.

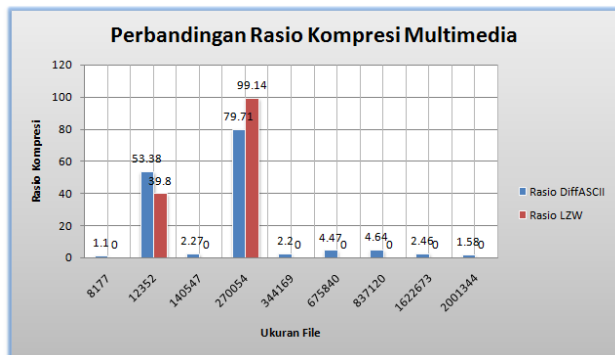
Tabel 5. Perbandingan Hasil Kompresi Multimedia

Nama File	Ukuran File	Kompresi	
		DiffASCII	LZW
Gambar1.jpg	12352	5758	7436
gambar2.bmp	270054	54791	2320
gambar3.png	8177	8087	7978
Audio1.mp3	1622673	1582685	1617155
Audio2.mp3	2001344	1969674	1998942
Audio3.wma	140547	137353	140237
Video1.avi	837120	798297	844037
Video3.3gp	344169	336588	401255
Video2.avi	675840	645629	708556

Tabel 6. Perbandingan Rasio Hasil Kompresi Multimedia

Nama File	Ukuran File	Kompresi	
		DiffASCII	LZW
Gambar1.jpg	12352	53.38	39.8
gambar2.bmp	270054	79.71	99.14
gambar3.png	8177	1.1	2.43

Audio1.mp3	1622673	2.46	0.34
Audio2.mp3	2001344	1.58	0.12
Audio3.wma	140547	2.27	0.22
Video1.avi	837120	4.64	0
Video3.3gp	344169	2.2	0
Video2.avi	675840	4.47	0



Gambar 4. Grafik Perbandingan Rasio Kompresi Multimedia

Perbandingan rasio kompresi pada jenis berkas multimedia yang telah dilakukan pada penelitian ini menunjukkan hasil yang cukup mengejutkan dimana hampir pada semua berkas multimedia yang dikompresi menggunakan metode LZW tidak dapat menghasilkan rasio kompresi yang baik. Sedangkan metode differensiasi ASCII walaupun rasio kompresi pada setiap berkas masukan relative kecil namun hasil keluaran kompresi selalu lebih kecil dibandingkan dengan ukuran aslinya. Adapun rendahnya rasio kompresi dari kedua metode disebabkan oleh seberapa besar berkas multimedia yang digunakan pada pengujian perbandingan ini adalah berkas dengan format yang telah terkompresi sejak awal seperti jpg, png, mp3, dan 3gp yang merupakan format *standard* multimedia terkompresi. Namun pada berkas multimedia yang secara format tidak terkompresi seperti bmp (Bitmap) baik metode differensiasi ASCII maupun LZW mampu mencapai rasio kompresi yang cukup tinggi.

4. Kesimpulan

Perbandingan rasio kompresi yang telah dilakukan pada penelitian ini yang membandingkan metode kompresi differensiasi ASCII dengan metode LZW memberikan hasil analisa yang cukup menarik. Differensiasi ASCII yang menggunakan konsep nilai selisih antar byte yang terdapat pada simbol masukkan memiliki nilai rasio kompresi yang cukup stabil. Rasio kompresi yang dihasilkan pada berkas teks secara rata-rata berada pada nilai 15 – 40%. Pada metode LZW rasio kompresi teks sangat bergantung pada isi dari teks masukan itu sendiri dimana jika teks masukan memiliki probabilitas

pasangan yang tinggi maka rasio kompresi yang dicapai juga sangat tinggi hingga mencapai nilai rasio sebesar 71%. Namun jika dokumen masukan berukuran kecil atau probabilitas pasangan simbol sangat rendah performa LZW pada kompresi teks sangat rendah dibandingkan dengan metode Differensiasi ASCII. Pada berkas multimedia, kedua metode secara umum memiliki performa rasio kompresi yang hampir sama, namun differensiasi ASCII memiliki rasio kompresi yang lebih baik dibandingkan dengan LZW walaupun dengan rasio kompresi yang cukup kecil dikarenakan sifatnya yang menggunakan nilai selisih sehingga tidak terlalu terpengaruh terhadap pola kemunculan dari simbol dari masukan. Kesimpulan yang diperoleh adalah nilai rasio yang paling baik jika dinilai dari besarnya adalah metode LZW sedangkan rasio kompresi yang lebih stabil dan mampu diterapkan pada hampir semua format adalah differensiasi ASCII.

References

- [1] Sharma, M. (2010). Compression Using Huffman Coding. IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5.
- [2] Schlosberg, A., Lam, B. Y., Yeo, G. S., & Clifton, R. J. (2013). DEFLATE Compression Algorithm Corrects for Overestimation of Phylogenetic Diversity by Grantham Approach to Single-Nucleotide Polymorphism Classification. International Journal of Molecular Sciences, 15, 8491-8508.
- [3] Tommy., Siregar, R., Lubis, I., E, Andi Marwan., A, Mahmud H., & Harahap, M. (2018). A Simple Compression Scheme Based on ASCII Value Differencing. IOP Conf. Series: Journal of Physics: Conf. Series 1007 (2018) 012022, doi :10.1088/1742-6596/1007/1/012022.
- [4] Al-laham, M., & El Emary, I. M. (2007). Comparative Study between Various Algorithms of Data Compression Techniques. IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.4, April 2007.
- [5] Sidhu, A. S., & Garg, E. M. (2014). Research Paper on Text Data Compression Algorithm using Hybrid Approach. IJCSMC, Vol. 3, Issue. 12, December 2014, pg.01 – 10, ISSN 2320-088X.
- [6] Sailunaz, K., Kotwal, M. R., & Huda, M. N. (2014). Data Compression Considering Text Files. International Journal of Computer Applications (0975 – 8887), Volume 90 – No 11, March 2014.
- [7] Brar, R. S., & Singh, B. (2013). A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data. International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013, ISSN: 2277 128X.

- [8] Porwal, S., Chaudhary, Y., Joshi, J., & Jain, M. (2013). Data Compression Methodologies for Lossless Data and Comparison between Algorithms. International Journal of Engineering Science and Innovative Technology (IJESIT), Volume 2, Issue 2, March 2013, ISSN: 2319-5967.
- [9] Singh, P. (2015). Lossless Data Compression Techniques And Comparison Between The Algorithms. International Research Journal of Engineering and Technology (IRJET), Volume: 02 Issue: 02, May-2015, e-ISSN: 2395-0056.
- [10] Alasmer, Z. M., Zahran, B. M., Ayyoub, B. A., & Kanan, M. A. (2013). A Comparison between English and Arabic Text Compression. Contemporary Engineering Sciences, Vol. 6, 2013, no. 3, 111 - 119, HIKARI Ltd.
- [11] Pratas, D., Silva, R. M., & Pinho, A. J. (2018). Comparison of Compression-Based Measures with Application to the Evolution of Primate Genomes. Entropy 2018, 20, 393; doi:10.3390/e20060393.
- [12] Yuan, X., & Haimi-Cohen, R. (2018). Image Compression Based on Compressive Sensing:End-to-End Comparison with JPEG. arXiv:1706.01000v2 [cs.CV] 28 Jul 2018.
- [13] Senthilkumaran, N., & Vinodhini, M. (2017). Comparison of Image Compression Techniques for MRI Brain Image. International Journal of Engineering Trends and Applications (IJETA) – Volume 4 Issue 2, Mar-Apr 2017, ISSN: 2393-9516.
- [14] AbuBaker, A., Eshtay, M., & AkhoZahia, M. (2016). Comparison Study of Different Lossy Compression Techniques Applied on Digital Mammogram Images. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 12, 2016.
- [15] M, Harahap., AM Husein., W, Purba, (2017). Metode N-Gram Untuk Ekstraksi Pola Teks Dalam Ukuran Besar Pada Algoritma Kompres Huffman Dan Lempel-Ziv-Welch (LZW) Yang Dimodifikasi. SNIKOM 14 September 2017. PP 180-186.
- [16] Mamun, M., & Sobhan, M. A. (2013). Hardware Approach of Lempel Ziv Welch Algorithm for Binary Data Compression. Bangi: IDOSI Publications.
- [17] Li, S., Li, C., & Kuo, J. (2011). On The Security Of A Secure Lempel Ziv Welch (LZW) Algorithm. Barcelona: IEEE.

Biografi Penulis



Tommy, Lahir di Medan, 13 November 1985. Pendidikan Strata 1 (S1) Program Studi Teknik Informatika di Sekolah Tinggi Teknik Harapan Medan yang sekarang telah menjadi Universitas

Program Studi Teknik Informatika
Universitas Prima Indonesia (UNPRI) Medan

Harapan Medan dan Strata 2 (S2) Magister Ilmu Komputer Program Studi Ilmu Komputer di Pascasarjana UPI YPTK Padang. Merupakan pengajar di Universitas Harapan Medan, Fakultas Teknologi Dan Komputer, Program Studi Teknik Informatika, Sumatera Utara. Minat penelitian saat ini meliputi bidang pattern recognition, artificial intelligence, data compression, dan image processing.



Rosyidah Siregar, Lahir di Medan, 7 Mei 1988. Pendidikan Strata 1 (S1) Program Studi Teknik Informatika di Sekolah Tinggi Teknik Harapan Medan yang sekarang telah menjadi Universitas Harapan Medan dan Strata 2 (S2) Magister Ilmu Komputer Program Studi Ilmu Komputer di Pascasarjana UPI YPTK Padang. Merupakan pengajar di Universitas Harapan Medan, Fakultas Teknologi Dan Komputer, Program Studi Teknik Informatika, Sumatera Utara.



Amir Mahmud Husein, memperoleh gelar Sarjana Komputer (S.Kom), Program Studi Sistem Informasi Sekolah Tinggi Teknik Harapan Medan yang sekarang telah menjadi Universitas Harapan Medan, lulus tahun 2008. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Ilmu Komputer Universitas Sumatera Utara, lulus tahun 2010. Saat ini aktif sebagai pengajar di Universitas Prima Indonesia Medan, Fakultas Teknologi dan Ilmu Komputer, Program Studi Teknik Informatika.



Mawaddah Harahap, memperoleh gelar Sarjana Komputer (S.Kom), Program Studi Sistem Informasi Sekolah Tinggi Teknik Harapan Medan yang sekarang telah menjadi Universitas Harapan Medan, lulus tahun 2008. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Ilmu Komputer Universitas Sumatera Utara, lulus tahun 2014. Merupakan pengajar di Universitas Prima Indonesia Medan, Fakultas Teknologi dan Ilmu Komputer, Program Studi Teknik Informatika, Sumatera Utara.



Ferdy Riza, Lahir di Kabupaten 50 Kota, 3 Juni 1989. Pendidikan Strata 1 (S1) Program Studi Teknik Informatika di Sekolah Tinggi Teknik Harapan dan Strata 2 (S2) Magister Ilmu Komputer Program Studi Teknik Informatika di

Pascasarjana Universitas Putra Indonesia. Saat ini mengajar di STMIK Logika dan aktif dalam bidang penelitian yang terkait dengan Kriptografi.